

第5章 数组

本章目标

- ◆ 理解数组概念及其在内存中的存储结构
- ◆ 一维数组定义、初始化、引用
- ◆ 二维数组定义、初始化、引用
- ◆ 字符数组及字符串
- ◆ 使用数组处理批量同类型数据问题
- ◆ 应用所学知识灵活处理数据存储问题

5.1 初识数组

◆ 例5-1 考试成绩统计，输入一组学生成绩以-1作为结束标记，计算所有成绩的平均分并统计高于平均分人数。

◆ 分析：

- 输入一组数据求平均分，使用循环就可以解决，读入数据累加，然后计算平均分（后读入数据覆盖先读入数据）
- 若要统计高于平均分人数，就需要遍历之前所有输入的成绩数据，但是之前所学知识无法解决这个问题。
- 使用数组存储读入的成绩数据，再求出平均分后，再遍历成绩数组，统计出高于平均分的人数。

例5-1 代码

```
#include <stdio.h>
int main()
{
    float sum=0,ave;
    int score;
    int scores[100];
    int num=0,count=0,i;
    printf("请输入学生成绩（以-1作为结束标记）：\n");
    scanf("%d",&score);
    while(score!=-1){
        scores[num]=score;
        sum+=score;
        num++;
        scanf("%d",&score);
    }

    if(num!=0){
        ave=sum/num;
        printf("学生平均成绩为：
%.1f\n",ave);
        for(i=0;i<num;i++){
            if(scores[i]>ave)
                count++;
        }
        printf("高于平均成绩的学生人数为
%d\n",count);
    }
    else
        printf("没有输入有效成绩！\n");
    return 0;
}
```

5.1.2 一维数组的定义及使用

◆ 一维数组的定义形式：

类型名 数组名[数组长度]

◆ 说明：

- 类型名确定每个数组元素的数据类型
- 数组名是数组变量的名称，需要满足标识符命名原则
- 数组长度，必须是确定的值

```
int num[5];
```

```
int n=5;
```

```
int num[n];
```

```
#define M 5
```

```
.....
```

```
int num[M];
```

数组在内存中的存储

| 内存地址 | | 数组下标 | 数组元素 |
|------------|-----|------|--------|
| num | | | |
| 0x0060FEB2 | 25 | 0 | num[0] |
| 0x0060FEB6 | 13 | 1 | num[1] |
| 0x0060FEBA | -12 | 2 | num[2] |
| 0x0060FEBE | 36 | 3 | num[3] |
| 0x0060FEC2 | 54 | 4 | num[4] |
| | | | |

数组的初始化

◆ 数组初始化格式

类型名 数组名[数组长度]={数组元素初值表};

◆ 例如：

```
int num[5]={25,13,-12,36,54};
```

```
int num[]={25,13,-12,36,54};
```

```
int num[5]={25,13};
```

◆ 说明：

- 定义同时为数组元素赋值
- 若对所有元素赋值，可以省略数组长度

数组元素的引用

◆ 数组元素引用：数组名[下标]

◆ 例如：num[i+1]=num[i]*2;

◆ 说明：

➤ 数组下标从0开始，可以使用的范围为0~数组长度-1

➤ 注意数组下标不能越界

➤ 数组定义完成后，只能通过下标对数组每个元素进行单独的使用，而不能单独使用数组名对整个数组进行访问

```
int num[5]; int i; for(i=0;i<5;i++)  
num[i]=i*2;
```


使用一维数组解决问题

1、遍历数组

【例 5-2】键盘输入一个班级人数 n ，以及 n 个学生《C语言程序设计》的分数，请编写程序对成绩进行分析，即统计最高分、最低分、平均分以及及格比例。

- 分析
 - n 个学生成绩需要记录，所以使用数组
 - 分析时，需要遍历所有学生成绩，通过数组下标的变化实现对数组元素的遍历
 - 程序分为三部分设计：数据输入、找最大最小、输出结果

示例代码

```
#include <stdio.h>
int main()
{
    int scores[50],max,min,sum=0;
    int num,i,count=0;
    double ave,rate;
    printf("请输入学生人数: ");
    scanf("%d",&num);
    printf("请输入学生成绩: \n");
    for(i=0;i<num;i++)
        scanf("%d",&scores[i]);
    max=scores[0];
    min=scores[0];
    for(i=0;i<num;i++){
        sum+=scores[i];
        if(max<scores[i])
            max=scores[i];
        if(min>scores[i])
            min=scores[i];
        if(scores[i]>=60)
            count++;
    }
}
```

```
/*平均分定义为double类型，计算时将sum
强制类型转换为double*/
    ave=(double)sum/num;
    /*计算及格率时，需要将count强制类型
转换为double类型*/
    rate=1.0*count/num;
    printf("*****成绩分析*****\n");
    printf("最高分是: %d, 最低分是:
%d\n",max,min);
    /*输出使用格式控制符%0.1lf指明输出保
留1位小数*/
    printf("平均分是: %0.1lf\n",ave);
    /*使用%%输出%，输出百分比结果因此输
出内容为rate*100*/
    printf("及格率: %.1lf%%",rate*100);
    return 0;
}
```

使用一维数组解决问题

2、查找数组

【例 5-3】有一门课程的考试成绩，按照学号顺序存在数组中，现在输入一个成绩，想知道考这个成绩的学生学号，该如何实现呢？

- 分析
 - 查找问题，遍历数组，找到后对应下标即为所求
 - 查找的两种情况：
 - 1、遍历数组，找到第一个就结束（从循环中break）
 - 2、遍历数组，要找到所有相等的数据（遍历所有元素，设置是否找到的标记）

示例代码

```
#include <stdio.h>
#define M 50
int main()
{
    int iScore[M],input;
    int iNum,i,flag=0;
    /*flag标记是否查找到，0表示没有查找到*/
    printf("请输入班级人数：");
    scanf("%d",&iNum);
    printf("请输入%d个成绩：\n",iNum);
    for(i=0;i<iNum;i++)
        scanf("%d",&iScore[i]);
    printf("请输入待查询成绩：");
    scanf("%d",&input);
```

```
    for(i=0;i<iNum;i++)
        if(input==iScore[i]){
            if(flag==0)
                printf("成绩为%d的学号为：
%d",input,i+1);
            /*如果是第一次找到则输出*/
            /*学号从1开始，而数组下标从0
开始，所以返回i+1*/
            else
                printf(" %d",i+1);
            flag=1;
        }
    if(flag==0)
        printf("没有找到成绩为%d的学生
",input);
    return 0;
}
```

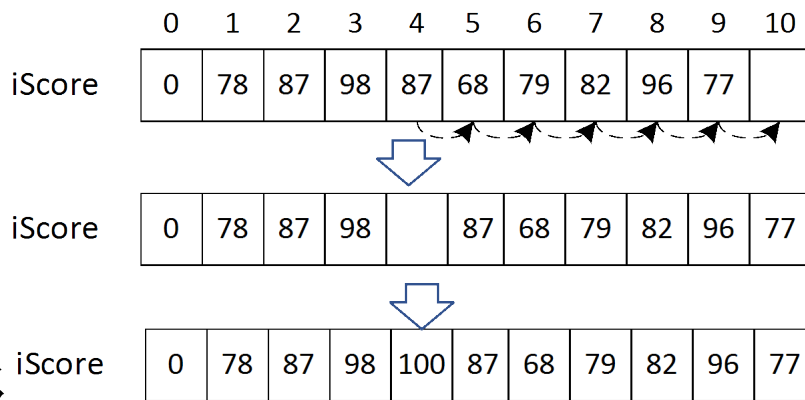
使用一维数组解决问题

3、数组中插入数据

【例 5-4】老师在录入成绩结束后核对发现，有一个同学的成绩落下了，导致从这个同学往后所有的同学成绩都向前串了一位。输入落后的同学的学号及对应的成绩，请编写程序帮助老师将成绩单整理正确。

- 分析

- 列表插入问题
- 不能直接在对应位置赋值
- 需要将后面的元素一次后移



示例代码

```
#include <stdio.h>
#define M 20
int main(){
    int
iScore[M]={0,78,87,98,87,68,79,82,96,77};
    /*为使学号与下标对应，成绩从第1个数
组元素开始存储*/
    int ild,i,input;
    printf("请输入插入成绩的学号：");
    scanf("%d",&ild);
    printf("请输入插入的成绩：");
    scanf("%d",&input);

    if(ild<1 || ild>10)
        printf("输入学号有误！\n");
    else{
        for(i=10;i>ild;i--)
            iScore[i]=iScore[i-1];
        iScore[i]=input;
    }
    printf("插入的成绩序列：");
    for(i=1;i<=10;i++)
        printf("%d ",iScore[i]);
    return 0;
}
```

使用一维数组解决问题

4、借助数组计数

【例 5-5】一门课程期末考试成绩出来后，老师想了解得分情况，请你编写程序统计一下每个分数成绩都有多少人。

- 分析
 - 要统计每个分数的人数，从第一个出现的成绩开始计数，会使程序十分复杂。
 - 百分制成绩数据范围一定在0~100之间，将成绩作为数组下标，数组元素值用来计数
 - `iCount[88]`对应的值为分数为88的人数。

示例代码

```
#include <stdio.h>
#define M 50
int main()
{
    int iScore[M],iNum,i;
    int iCount[101]={0};
    /*使用iCount[i]记录成绩为i的学生人数*/
    printf("请输入总人数: ");
    scanf("%d",&iNum);
    printf("请输入%d个成绩: \n",iNum);
    for(i=0;i<iNum;i++)
        scanf("%d",&iScore[i]);
    for(i=0;i<iNum;i++)
        /*成绩iScore[i]对应下标的iCount数组
元素累加*/
        iCount[iScore[i]]++;
    for(i=100;i>=0;i--)
        if(iCount[i]!=0)
            printf("%d分%d人\n",i,iCount[i]);
    return 0;
}
```


使用一维数组解决问题

5、排序问题

【例 5-6】期末考试结束后，若需要对一个班级学生的《C语言程序设计》成绩进行排序，应该如何完成呢？

排序问题，先将问题进行分解：

1. 从第0个元素开始，找到数组的最大值的下标
2. 将最大值与第0个元素互换
3. 从第i个元素开始向后，找到最大值的下标，并且将最大值与第i个元素互换
4. i从0到n-1，重复第3个步骤实现排序（选择法排序）

从第0个元素开始，找到数组的最大值的下标

k=0
max=78
maxIndex=0

| | | | | | | |
|----|----|----|----|----|----|--|
| | k | | | | | |
| | ↓ | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | |
| 78 | 94 | 82 | 63 | 98 | 84 | |

k=1
max=94
maxIndex=1

| | | | | | | |
|----|----|----|----|----|----|--|
| | | k | | | | |
| | | ↓ | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | |
| 78 | 94 | 82 | 63 | 98 | 84 | |

k=2
max=94
maxIndex=1

| | | | | | | |
|----|----|----|----|----|----|--|
| | | | k | | | |
| | | | ↓ | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | |
| 78 | 94 | 82 | 63 | 98 | 84 | |

k=3
max=94
maxIndex=1

| | | | | | | |
|----|----|----|----|----|----|--|
| | | | | k | | |
| | | | | ↓ | | |
| 0 | 1 | 2 | 3 | 4 | 5 | |
| 78 | 94 | 82 | 63 | 98 | 84 | |

k=4
max=98
maxIndex=4

| | | | | | | |
|----|----|----|----|----|----|--|
| | | | | | k | |
| | | | | | ↓ | |
| 0 | 1 | 2 | 3 | 4 | 5 | |
| 78 | 94 | 82 | 63 | 98 | 84 | |

k=5
max=98
maxIndex=4

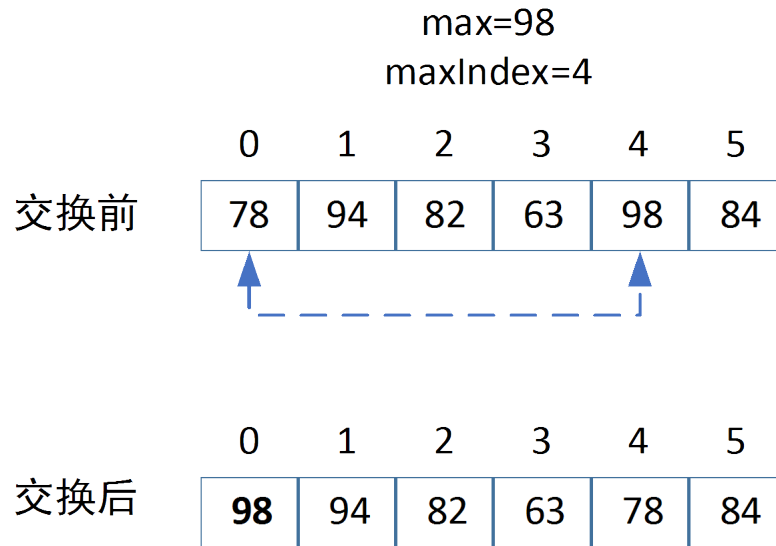
| | | | | | | |
|----|----|----|----|----|----|---|
| | | | | | | k |
| | | | | | | ↓ |
| 0 | 1 | 2 | 3 | 4 | 5 | |
| 78 | 94 | 82 | 63 | 98 | 84 | |

从第0个元素开始，找到数组的最大值的下标

```
#include <stdio.h>
#define M 50
int main()
{
    int iScore[M],iNum,k,maxIndex;
    printf("请输入总人数: ");
    scanf("%d",&iNum);
    printf("请输入%d个成绩: \n",iNum);
    for(k=0;k<iNum;k++)
        scanf("%d",&iScore[k]);
    /*使用maxIndex记录最大值下标，iScore[maxIndex]为当前最大值
    初始假设iScore[0]为最大值，因此初始时maxIndex=0*/
    maxIndex=0;
    for(k=1;k<iNum;k++)
        /*遍历数组，若记录的最大值比当前元素值小，则更新最大值下标*/
        if(iScore[maxIndex]<iScore[k])
            maxIndex=k;
    printf("成绩最大值为%d，下标为%d\n",iScore[maxIndex],maxIndex);
    return 0;
}
```

将最大值与第0个元素互换

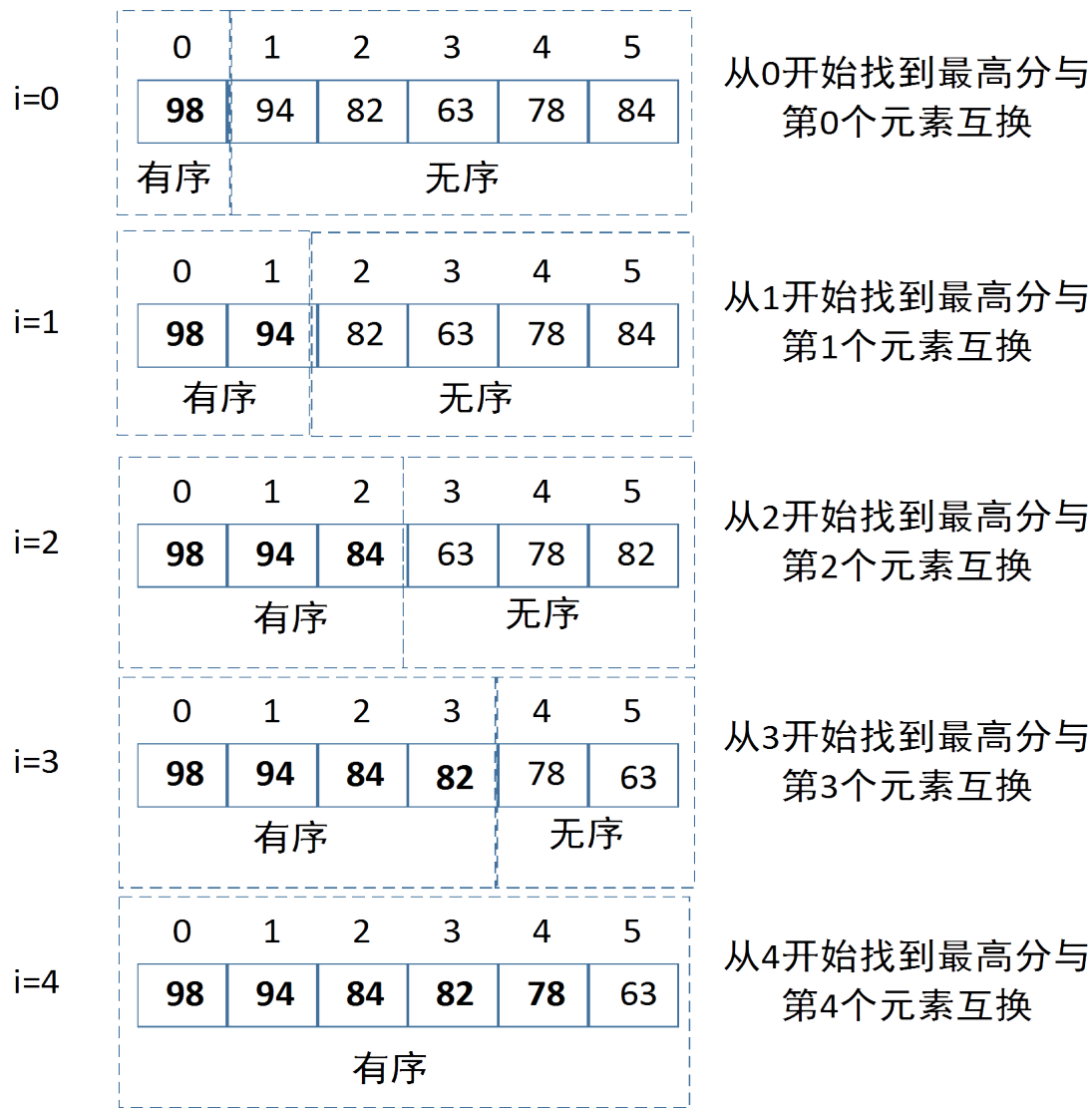
```
int tmp; tmp=iScore[0];  
iScore[0]=iScore[maxIndex];  
iScore[maxIndex]=tmp;
```



从第i个元素开始向后，找到最大值的下标，并且将最大值与第i个元素互换

```
maxIndex=i; for(k=i+1;k<iNum;k++) /*遍历数组，若记录的最大值比当前元素值小，则更新最大值下标*/ if(iScore[maxIndex]<iScore[k])
maxIndex=k;    tmp=iScore[i]; iScore[i]=iScore[maxIndex];
iScore[maxIndex]=tmp;
```

i从0到n-1，重复第3个步骤实现排序（选择法排序）



i从0到n-1，重复第3个步骤实现排序（选择法排序）

```
1  for(i=0;i<iNum-1;i++)
2  {
3      maxIndex=i;
4      for(k=i+1;k<iNum;k++)
5          if(iScore[maxIndex]<iScore[k])
6              maxIndex=k;
7      tmp=iScore[i];
8      iScore[i]=iScore[maxIndex];
9      iScore[maxIndex]=tmp;
10 }
```

外层循环

找当前最大值
(这里判断的是排序依据)

将找到最大值与最前面元素互换
(这里互换的是排序的对象)

使用一维数组解决问题

6、借助数组使程序易于维护

【例 5-7】学校每学期都会进行奖学金的评定，综合奖学金的评定成绩由以下几部分构成：学业平均成绩80%，竞赛获奖10%，班级贡献5%，活动参与5%，现在已知一位同学每部分百分制成绩，使用数组存储，请计算他的综合评定成绩。要考虑程序的可维护性。

- 软件维护是指在软件产品发布后，因修正错误、提升性能或其他属性而进行的软件修改
- 为了便于维护，将每个分项成绩的占比存放在数组中，将来修改更方便

示例代码

```
#include <stdio.h>
#define M 5
int main()
{
    int iScore[M]={0};
    int iRate[M]={0,80,10,5,5};
    /*分项成绩存储在iScore[1]~iScore[4]中，综合成绩计算完成存入iScore[0]*/
    int i;
    printf("请输入学业平均成绩、竞赛获奖、班级贡献和活动参与成绩：\n");
    for(i=1;i<M;i++)
        scanf("%d",&iScore[i]);
    for(i=1;i<M;i++)
        iScore[0]+=iScore[i]*iRate[i];
    iScore[0]/=100;
    printf("该同学综合成绩是： %d\n",iScore[0]);
    return 0;
}
```

5.2 二维数组

【例5-8】输入一个专业所有学生的学业平均成绩、竞赛获奖成绩、班级贡献成绩、活动参与成绩，并且按照学业平均成绩80%，竞赛获奖10%，班级贡献5%，活动参与5%来计算每个学生的综合成绩，请编写程序完成这个任务。

二维数组解决该问题

- ◆ 每个学生有四项成绩，需要使用数组记录（一个维度）
- ◆ 多个学生构成另一个维度

```

#include <stdio.h>
#define MAXSTUNUM 50
#define M 5
int main()
{
    int iScore[MAXSTUNUM][M]={0};
    int iRate[5]={0,80,10,5,5};
    /*分项成绩存储在iScore[1]~iScore[4]中,
    综合成绩计算完成存入iScore[0]*/
    int i,j,iNum;
    printf("请输入学生人数: ");
    scanf("%d",&iNum);
    printf("请输入每位学生的学业平均成绩、
    竞赛获奖、班级贡献和活动参与成绩:
    \n");
    for(j=0;j<iNum;j++)
        for(i=1;i<M;i++)
            scanf("%d",&iScore[j][i]);

    for(j=0;j<iNum;j++){
        for(i=1;i<M;i++)
            iScore[j][0]+=iScore[j][i]*iRate[i];
        iScore[j][0]/=100;
    }
    for(j=0;j<iNum;j++)
        printf("第%d位同学的综合成绩是:
        %d\n",j+1,iScore[j][0]);
    return 0;
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/467112166043006151>